



**University of
Zurich^{UZH}**

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2015

FlexiView: A Magnet-Based Approach for Visualizing Requirements Artifacts

Ghazi, Parisa ; Seyff, Norbert ; Glinz, Martin

Abstract: Requirements engineers create large numbers of artifacts when eliciting and documenting requirements. They need to navigate through these artifacts and display information details at points of interest for reviewing or editing information. [Question/problem] Traditional visualization mechanisms such as scrolling and opening multiple windows lose context when navigating and can be cumbersome to use, hence. On the other hand, focus+context approaches can display details in context, but they distort the data shown (e.g., fisheye views) or result in a large display canvas which again requires scrolling (e.g., zooming in ADORA). [Principal ideas/results] We are developing a novel method for displaying just the information needed to perform an intended task. Our method partitions the available screen space into regions. The boundaries of regions are simulated with a model consisting of virtual magnetic balls and springs that behaves like a physical system. This model supports the requirements engineer in selecting how the relevant information should be displayed. [Contribution] In this paper, we present preliminary results on how our conceptual solution works and what benefits are expected.

DOI: https://doi.org/10.1007/978-3-319-16101-3_18

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-110728>

Conference or Workshop Item

Accepted Version

Originally published at:

Ghazi, Parisa; Seyff, Norbert; Glinz, Martin (2015). FlexiView: A Magnet-Based Approach for Visualizing Requirements Artifacts. In: 21st International Working Conference on Requirements Engineering: Foundation for Software Quality , REFSQ 2015, Essen, Germany, 23 March 2015 - 26 March 2015, Springer International Publishing.

DOI: https://doi.org/10.1007/978-3-319-16101-3_18

FlexiView: A Magnet-Based Approach for Visualizing Requirements Artifacts

Parisa Ghazi, Norbert Seyff and Martin Glinz

Department of Informatics, University of Zurich, Switzerland
{ghazi,seyff,glinz}@ifi.uzh.ch

Abstract. [Context and motivation] Requirements engineers create large numbers of artifacts when eliciting and documenting requirements. They need to navigate through these artifacts and display information details at points of interest for reviewing or editing information. [Question/problem] Traditional visualization mechanisms such as scrolling and opening multiple windows lose context when navigating and can be cumbersome to use, hence. On the other hand, focus+context approaches can display details in context, but they distort the data shown (e.g., fisheye views) or result in a large display canvas which again requires scrolling (e.g., zooming in ADORA). [Principal ideas/results] We are developing a novel method for displaying just the information needed to perform an intended task. Our method partitions the available screen space into regions. The boundaries of regions are simulated with a model consisting of virtual magnetic balls and springs that behaves like a physical system. This model supports the requirements engineer in selecting how the relevant information should be displayed. [Contribution] In this paper, we present preliminary results on how our conceptual solution works and what benefits are expected.

Keywords: Requirements Engineering, visualization, focus+context, physics-based interface, magnet

1 Introduction

When eliciting and documenting requirements, requirements engineers create a large number of artifacts (e.g., documents, models, or sketches). Creating and working with these artifacts on electronic devices entails two visualization problems, particularly when working with displays of limited size (e.g., tablets): (i) There are artifacts such as large models or sketches that are larger than the available display. (ii) A requirements engineer frequently needs to view more than one artifact concurrently in order to comprehend or edit these artifacts.

Today's tools employ traditional techniques for tackling these visualization problems: the first problem is typically addressed by scrolling and the second one by opening multiple windows [1]. These techniques work well for focusing on individual pieces of information, but they do this at the expense of losing the information about the context that those pieces are embedded in. Therefore, working with traditional visualization mechanisms is cumbersome when the elements to be displayed in detail

are part of a network of interconnected elements, which is typically the case in Requirements Engineering (RE). On the other hand, there are so-called focus+context visualization approaches that can display details in context [1], [6]. However, the existing approaches distort the data shown (e.g., fisheye views) [4], [10] or result in a large display canvas which requires scrolling (e.g., zooming in ADORA) [8].

In our research we are developing a new visualization mechanism called *FlexiView* which solves, in a unified way, both visualization problems mentioned above. Based on a *physical metaphor of magnets and springs* [2], [9], [11], FlexiView shall be able to flexibly visualize detailed requirements artifacts without losing the surrounding context within a display canvas of fixed size. In contrast to existing visualization mechanisms, FlexiView will be designed such that it can be used for visualizing both single artifacts (e.g., a graphic model diagram or a sketch) and a network of multiple different artifacts.

The rest of the paper is organized as follows. In Section 2, we briefly discuss the goals of our approach. Section 3 reviews the relevant literature. In Section 4, we present our approach and discuss its features and benefits. Section 5 concludes the paper.

2 Research Goals

Our goal is to develop a unified focus+context visualization mechanism which is tailored to requirements engineering. With our approach, we aim at overcoming the problems of existing visualization approaches for RE artifacts, thus allowing the construction of innovative RE tools (e.g., for supporting lightweight requirements modeling [5]) as well as improving the way how existing RE tools visualize information. We envisage that such tools will (i) reduce the time and energy spent on navigating among various artifacts, (ii) prevent users (requirements engineers as well as stakeholders and developers) from getting lost in the navigation space, and (iii) make the set of RE artifacts better comprehensible for users. We expect that our visualization mechanisms will be useful also for visualizing other artifacts, e.g., in software architecture, but we will concentrate on RE artifacts in our research.

3 Related Work

Scrolling and opening multiple windows are traditional ways to deal with a large number of artifacts. They have been used in almost all available user interfaces. However, they lose context and create visual discontinuities, thus causing cognitive overhead for the user [1].

Focus+context visualization techniques display the focus within its context in a single continuous view. The theoretical foundation for focus+context interfaces was established by Furnas [4], who describes generalized fisheye views. This is a general interaction framework for information filtering according to the user's current point of interest. This concept was later used for creating Graphical Fisheye Views (GFV) [10]. GFV is a non-linear distortion-oriented graphical visualization technique and supports multiple foci. The results are sometimes reported as too distorted. Many derivations of fisheye views can be found in literature, such as JellyLens [7], that

morphs around arbitrary geometric features in the data. In the ADORA project, a fisheye zoom algorithm for visualizing and manipulating hierarchical graphical ADORA models was developed [8]. The algorithm provides an editable layout which is stable under multiple zooming operations. However, zooming in multiple points may result in a large canvas which requires the user to scroll again.

In the field of graph visualization, many techniques and algorithms have been created for viewing large graphs. A particular thread of work deals with manipulating graph visualizations based on a physical metaphor [9], treating graph nodes as metal balls and edges as springs that are flexibly attached to those balls [2]. By applying forces to such a network of balls and springs, for example by placing magnets, interesting parts of a graph can be highlighted or magnified [11], thus allowing the construction of intuitive, user-friendly graph visualization and navigation mechanisms.

4 FlexiView: A Magnet-Based Visualization Approach

FlexiView combines the concepts of fisheye zooming and magnet-based graph visualization into a new technique for visualizing and manipulating requirements artifacts. We have chosen this technique due to its potential for solving both visualization problems mentioned in Sect. 1 (visualizing large individual artifacts as well as sets of interconnected artifacts) in a uniform way on display devices of limited size. Subsequently, we illustrate the idea using a typical scenario occurring in early stages of requirements engineering: we have a set of interconnected artifacts, each artifact being a chunk of text, a sketch, a model fragment, an image, etc.

4.1 Conceptual Solution

FlexiView partitions the whole working space into regions in such a way that each region contains just one element (i.e., a single artifact in the scenario mentioned above). For the sake of simplicity, we will call these elements objects. Unlike other visualization techniques, users interact with regions instead of objects. The interactions of the users affect the regions and any change in the regions affects the objects consequently. To manipulate the size of the regions, we model the region boundaries with a physical spring model [2] (Fig. 1a) and apply forces to that model using virtual magnets [11] (Fig. 1b). The four balls in the corners of the display space are consid-

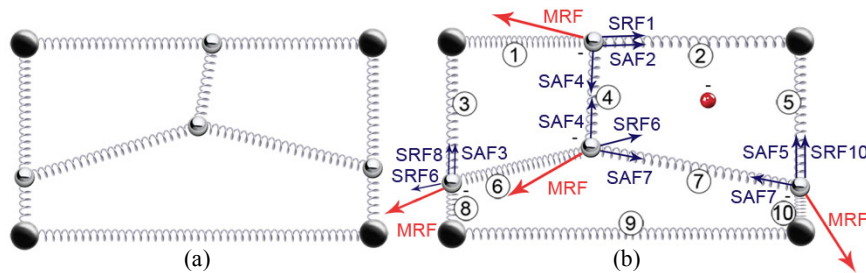


Fig.1. (a) A sample of regions modeled by metal balls and springs. (b) The positions of the balls are determined by three forces: the Spring Repulsive Force (SRF), the Spring Attractive Force (SAF), and the Magnet Repulsive Force (MRF).

ered to be fixed and neutral. All other balls can move and are considered to be magnetic, having a negative pole on their surface. The balls positioned on a horizontal or vertical edge of the display space can only move horizontally or vertically, respectively. The other balls can move in any direction.

In its initial position, the model is in balance. Users can now manipulate the size of regions by creating virtual magnets anywhere on the screen. These magnets have a single pole on their surface. The position, strength and polarity of these magnets determine how the regions change: any magnet repels the balls of the same polarity and attracts the balls of the opposite polarity. The placement of virtual magnets on the drawing space applies forces to the movable balls and makes them move, thus compressing or stretching the springs attached to the balls. Springs apply forces to the balls in return. The balls move until the forces of springs and the magnet(s) applied to them neutralize each other. The system is in balance again until the user changes the layout by creating or removing a magnet, moving an existing magnet, or altering its strength. Creating multiple magnets affects multiple regions simultaneously.

Figure 1b shows the forces and the resulting repositioning of balls when a magnet of negative polarity is placed in the top right region. Compressed springs apply repulsive forces and stretched springs apply attractive forces. The directions of spring forces are aligned with the directions of the springs. The direction of the force that the magnet applies is given by the straight line between the magnet and the ball. The balls on the boundaries of the drawing space behave in a restricted way as described above.

The size and the position of the objects are controlled by the regions they reside in. When the position or the size of a region is changed by the user, the new position or size of the objects residing in that region will be calculated accordingly. The result will be the enlargement or shrinkage of objects. Eventually a new view of the original information is produced. Figure 2 shows three steps of a user interaction. The first

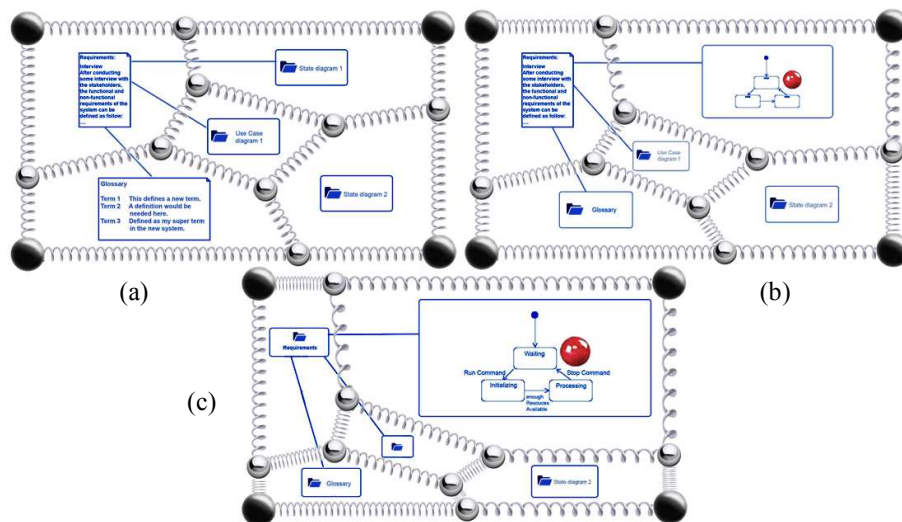


Fig.2. (a) RE artifacts and their regions. (b) The user has placed a magnet in the top right region, resulting in the enlargement of this region and the appearance of more details. In the shrunk region at the bottom left, fewer details are displayed. (c) The user has increased the strength of the magnet, so the the corresponding region grows and the other ones shrink.

image (2a) shows some objects representing requirements artifacts, their relations and their regions. The regions are modeled by our balls and springs model. In the second image (2b), the user has created a virtual magnet with negative polarity (the red ball) in the region of interest. The magnet has repelled the balls and caused the region of interest to increase in size. The object in this region is enlarged and can be displayed with more detail, hence. Conversely, the bottom left region has become too small to display its object in detail, so this object is replaced by a more abstract representation. In Figure 2c the user has increased the power of the magnet, resulting in a larger region of interest and further shrinkage of the other regions.

In order to replace objects in shrunk regions with more abstract representations and those in enlarged regions with more detailed ones, we keep display metadata for all objects [3]. We assume that we have at least a three-level hierarchy: project – artifact – contents of artifact. If an artifact, for example, is a symbol-and-line drawing, the symbols in that drawing constitute another level of hierarchy.

The applications used to create and edit artifacts store them in their own file format on local or remote storages or in repositories. We assume that these applications provide a kind of plug-in of FlexiView such that FlexiView can access the information required to display the artifacts and/or their constituents. Thus, users can explore information by navigating in and between artifacts with FlexiView while they can still manipulate and modify the content shown using the corresponding applications.

4.2 Algorithms for FlexiView

We are currently exploring existing graph manipulation algorithms that can be adapted for implementing the FlexiView approach. As in other work [11], we do not strive for physical accuracy, modeling exactly Hooke’s law for the springs and the laws of magnetism for the magnets, but use the physical model as a *metaphor* for guiding algorithm design. The users of FlexiView will not have to bother with physics. For them, using a magnet will feel like having a wizard that magnifies the region of interest on the display by a user-controlled factor and shrinks the rest accordingly.

4.3 Expected Benefits

Keeping the overview. A strong magnet can enlarge a region up to almost the whole working space and consequently shrink the other regions and their residing objects down to almost a dot. However the overview still exists. Although the undersized objects may be unclear, showing their relations and their positions keeps the complete image of the information in the user’s mind.

Minimizing distortion. All focus+context techniques distort the image of the information. In FlexiView the information inside each region alone is not distorted. The overall distortion available gradually increases when moving away from the current foci and decreases reaching far regions. Furthermore, the neighboring structure and relative position of the regions is kept intact. This way, the user is still capable of mapping the produced view to the original one, thus causing less disorientation.

Editing ability. Distorted views may improve the visualization, but are not pleasant when it comes to editing tasks. In our approach each region acts as an undistorted drawing canvas which enables users to edit information conveniently.

Being reversible. The altered views of the information are temporary views which are produced during specific tasks. The benefit of using magnets as tools of interaction is that by removing them, the original view reappears on the screen immediately. Moreover, the sequence of creating magnets on the screen can be undone not only in the reverse order but any magnet can be removed regardless of existing magnets created after it.

4.4 An Application Scenario

We illustrate the expected benefits of FlexiView with an application scenario from RE. Imagine a requirements engineer works on a requirements change request concerning the behavior of a component X. Let's follow this engineer's work through a sequence of steps. (1) The engineer starts from an overview that displays an interconnected set of requirements artifacts. (2) She places a magnet on the component X icon so that the constituents of component X appear. (3) She then places the magnet on the state machine icon of component X and increases the strength of the magnet until the state diagram appears. (4) Now she can study this diagram and figure out how it would be impacted by the requested change. (5) Next she wants to know the corresponding stakeholders. Placing another magnet on the pre-tracing link, she follows that link to the list of stakeholders, where she intensifies the strength of this magnet to see the actual stakeholders for the state machine of component X (the size of the state machine will shrink when displaying the stakeholder list, but it will remain a focus area on the display as its magnet is still there). (6) For a critical stakeholder, the engineer now wants to view this stakeholder's business goals. She moves the second magnet from the stakeholder list to the business goal specification, following the corresponding link. The stakeholder list disappears as soon as the magnet is moved and the region containing the business goal specification is enlarged. (7) By controlling the intensity of the magnet, she can now navigate into the business goals. (8) Having studied this information, she now wants to modify the state machine of component X. As the magnet on the state machine of component X is still there, she just removes the magnet from the business goal specification and the display reverts exactly to the situation that she had in step (4), thus allowing her to make the intended modification.

4.5 Research Status

We started this research in spring 2014 with conducting a thorough literature review. Based on the results of this review as well as an analysis of navigation and visualization problems identified in our FlexiSketch project [12], we have developed the concepts of FlexiView as a new technique for visualizing and manipulating requirements artifacts. We are currently investigating algorithms for implementing our approach. Our research will continue with actually implementing FlexiView and creating a test environment which will allow us to evaluate our approach against other approaches for visualizing and editing a set of requirements artifacts. We will evaluate the usefulness of FlexiView for performing typical RE tasks such as creating and understanding artifacts, tracing and change management. Additionally, we will deploy our approach on FlexiSketch [12], where we plan to conduct real-world evaluation studies.

5 Conclusions

In this paper we have previewed FlexiView: a novel visualization technique which aims at enabling requirements engineers to work with multiple interconnected artifacts on screens of limited size and, using the very same visualization technique, enabling them to navigate in artifacts that are larger than the available screen. Based on its underlying physical metaphor of springs and magnets, we expect FlexiView to provide seamless and natural looking multi-focus zooming. Due to its generic nature, FlexiView will be embeddable in both existing and novel tools that manipulate requirements artifacts such that these tools deliver their services through the FlexiView visualization mechanisms.

References

1. Cockburn, A., Karlson, A., Bederson, B.B.: A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, vol. 41, no. 1, pp. 1–31 (2008)
2. Eades, P.: A heuristic for graph drawing. *Congresses Numerantium*, vol. 42, pp.149–160. Utilitas Mathematica Publishing Inc., Winnipeg (1984)
3. Frisch, M., Dachsel, R., Brückmann, T.: Towards seamless semantic zooming techniques for UML diagrams. In: 4th ACM Symposium on Software Visualization (SoftVis’08), pp. 207–208 (2008)
4. Furnas, G.W.: Generalized fisheye views. In: SIGCHI Conference on Human Factors in Computing Systems (CHI’86), pp.16–23 (1986)
5. Glinz, M.: Very lightweight requirements modeling. In: 18th IEEE International Requirements Engineering Conference (RE’10), pp. 385–386 (2010)
6. Kagdi, H., Maletic, J.I.: Onion graphs for focus+context views of UML class diagrams. In: 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISOFT’07), pp. 80–87 (2007)
7. Pindat, C., Pietriga, E., Chapuis, O., Puech, C.: JellyLens: Content-aware adaptive lenses. In: 25th annual ACM Symposium on User Interface Software and Technology (UIST’12), pp. 261–270 (2012)
8. Reinhard, T., Meier, S., Stoiber, R., Cramer C., Glinz, M.: Tool support for the navigation in graphical models, In: 30th International Conference on Software Engineering (ICSE’08), pp. 823–826 (2008)
9. Rzeszotarski, J.M., Kittur, A.: Kinetica: Naturalistic multi-touch data visualization. In: SIGCHI Conference on Human Factors in Computing Systems (CHI’14), pp. 897–906 (2014)
10. Sarkar, M., Brown, M.H.: Graphical fisheye views. *Communications of the ACM*, vol. 37, no. 12, pp. 73–84 (1994)
11. Spritzer, A.S., Freitas C.M.D.S.: A physics-based approach for interactive manipulation of graph visualizations. In: Working Conference Advanced Visual Interfaces (AVI’08), pp. 271–278 (2008)
12. Wüest, D., Seyff, N., Glinz, M.: Semi-automatic generation of metamodels from model sketches. In: 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 664–669 (2013)